УДК 517.97

Н.С. Васильев

МНОГОПРОЦЕССОРНЫЕ СЕТИ: ПРОЕКТИРОВАНИЕ И УПРАВЛЕНИЕ НА ОСНОВЕ МНОГОКРИТЕРИАЛЬНОГО ПОДХОДА

Рассмотрена проблема проектирования многопроцессорной системы управления сложным техническим комплексом. Предложены архитектура и математическая модель функционирования сети. Построен эффективный алгоритм управления сетевыми вычислениями, учитывающий многоцелевое назначение системы.

E-mail: nik8519@yandex.ru

Ключевые слова: *параллельный алгоритм, граф алгоритма, бинарное отношение предшествования, цепь, многокритериальная оптимизация, маршрутизация, задача о назначениях.*

Сети процессорных устройств широко используют для оперативного управления техническими комплексами, функционирующими в сложной окружающей обстановке. Подобные системы имеют многоцелевое назначение. Поэтому их адекватное теоретическое описание дается с помощью многокритериальных моделей исследования операций.

В системах радиолокации, например, необходимо своевременно обнаруживать, идентифицировать, сопровождать, взаимодействовать и управлять одновременно большим количеством воздушных и наземных объектов. Все эти задачи тесно взаимосвязаны и могут быть решены при одних и тех же исходных данных. Качество работы всей системы оценивают по многим критериям, в том числе по надежности и быстродействию.

Математическое моделирование позволяет генерировать и подвергать анализу многие варианты проектных решений, связанных с созданием сети электронных устройств, управляющих работой технического комплекса. Сетевая архитектура позволяет распараллеливать процессы сбора и обработки информации с целью ускорения процесса решения задач. На этапе проектирования обеспечивают необходимую степень надежности и требуемое быстродействие сетевой архитектуры. Для этого применяют дублирование функциональных элементов и линий связи, а управление многопроцессорной системой осуществляют параллельными алгоритмами. Помимо качества системы при разработке принимают также во внимание дополнительный критерий — стоимость.

Выбор архитектуры сети. Поиск эффективных вариантов архитектуры и параметров сети должен учитывать многокритериальность целей управления и многозадачный режим использования разрабатываемой системы [1, 2]. Оценку качества работы системы можно проводить с помощью функциональной модели сети, включающей алгоритм управления сетевыми вычислениями.

Возможные варианты многопроцессорной сети зачастую формируют с помощью «сборки» графов алгоритмов решаемых задач. Для повышения надежности дополнительно проводят дублирование элементов сети. При этом подходе утрачивается универсальность системы, затруднены развитие сети и модификация применяемых алгоритмов.

Представляется целесообразным использовать многопроцессорные системы ярусно-параллельной архитектуры (ЯП-формы) достаточно большой ширины и высоты. Эти структуры надежны, просты для дальнейшего развития, а регулярность архитектуры облегчает реализацию быстрых параллельных алгоритмов решения поступающих задач.

Приведем некоторые определения [3]. Ярусами называются подмножества V_h, образующие разбиение множества всех узлов сети. Ориентированный граф сети имеет ЯП-форму, если всякая дуга графа инцидентна лишь узлам соседних ярусов с номерами h, $h + 1$, и имеет направление из узлов яруса V_h в узлы яруса V_{h+1}.

На множестве ярусов можно задавать бинарное отношение предшествования, соответствующее направлению дуг графа сети и позволяющее рассматривать ярусы верхнего уровня с более высоким номером. Под высотой ЯП-формы понимается число ярусов, а под шириной яруса — число элементов в нем.

Выполняемые в многопроцессорной системе, имеющей ЯП-форму, алгоритмы также должны быть представлены в виде параллельных программ, а все вычисления распределены между процессорами, размещенными в узлах этой сетевой архитектуры. Передача промежуточных данных происходит по дугам. Таким образом, ЯП-форма определяет упорядоченность вычислений в сети.

В сетевых узлах каждого яруса размещены электронные устройства одинаковой специализации. В памяти этих процессорных единиц хранятся программы выполнения заданий одного вида. Результаты работы программ выполнения заданий, загруженных в сетевые узлы, передаются по дугам сети, переходя с яруса на ярус. Полная связность сети обеспечивается посредством маршрутов, по которым поступают данные от произвольных узлов нижних ярусов к любым узлам верхних ярусов. Промежуточные сетевые узлы используются как транзитные.

Высота, ширина, набор дуг ЯП-формы, тип процессорных устройств, объемы памяти и другие параметры подлежат нахождению в процессе проектирования системы. Выбирают их в результате решения некоторой многокритериальной задачи, оценивающей качество работы всей системы. Например, это может быть стоимость системы, а также время решения каждого типа задач, поток которых поступает в сеть. Процесс проектирования специализированной архитектуры автоматизируют [1, 2], проводя оптимизацию системы путем поиска эффективного по Парето набора ее параметров [6].

Заметим, что для обработки заданий в универсальной суперкомпьютерной системе представление алгоритма в ярусно-параллельной форме также отвечает процессу распараллеливания вычислений.

Поток задач. Вычисления в сети. Технические системы функционируют во времени. Поэтому на вход сети поступают потоки однотипных задач, решаемых, возможно, с использованием результатов предыдущих вычислений.

Пусть имеется некоторый набор задач $k = 1, 2, \ldots, K$. Для сетевого решения нужно предварительно выявить параллелизм, присущий каждой задаче, и построить граф алгоритма ее решения [3]. Этот граф ориентирован. Вершинами представлены более простые задания «подходящего» уровня сложности, на которые разбита исходная задача. Задания связаны отношением предшествования, представленного дугами графа алгоритма: порядок выполнения заданий регламентируется в зависимости от результатов промежуточных вычислений. Выполнение всех заданий в соответствии с графом алгоритма дает искомое решение соответствующей задачи.

Пусть графы алгоритмов всех задач в сети $k = 1, 2, \ldots, K$ представлены в ЯП-форме. Для решения любой задачи спроецируем ее граф алгоритма в граф сети. При этом все вершины графа алгоритма переходят в узлы графа сети надлежащего уровня. Дуги графа алгоритма отображаются в маршруты, которые соединяют сетевые узлы, являющиеся проекциями вершин. Из заданий, спроецированных в один узел сети, формируется очередь на выполнение. (Проецирование можно проводить динамически, т. е. в процессе вычислений.)

Управление процессом вычислений в сети указанной архитектуры осуществляется с помощью потока данных — результатов выполнения предшествующих заданий. Промежуточные данные передаются по сети в соответствии с графами алгоритмов. По мере получения необходимых данных ожидающие их задания переходят в состояние готовности к выполнению (активизируются). Таким образом, на множестве всех заданий (программ), распределенных по сети, определено бинарное отношение предшествования.

По этому отношению на множестве всех заданий алгоритма выделим минимальные и максимальные элементы. Выполнение алгоритмов начинается с заданий, являющихся минимальными элементами, которые должны быть размещены в сетевых узлах нижних ярусов. Завершается решение задач в узлах верхних ярусов, где происходит выполнение заданий, являющихся максимальными элементами. В процессе работы системы источниками передаваемых данных становятся узлы, выполняющие задания, а стоками — узлы с заданиями, ожидающими данные в соответствии с графами алгоритмов. Во всех сетевых узлах активные задания выполняются в порядке установленных очередей.

ЯП-архитектура исключает возможность зацикливаний, связанных с бесконечными блужданиями данных по сети. Процессы передачи данных (будем называть их сообщениями) также допускают распараллеливание. Каждое сообщение можно разбивать на блоки (пакеты), снабженные нумерацией и метками, идентифицирующими решаемую задачу. Это потребуется для сборки переданного сообщения из пакетов в узле-адресате при многозадачном режиме работы сети. (При наличии длинных сообщений пакетная обработка ускоряет работу сети.)

Для алгоритмической реализации предложенной схемы вычислений необходимо решить многокритериальную задачу о назначениях, эквивалентную проецированию вершин графов алгоритмов в узлы графа сети, а также многокритериальную задачу маршрутизации потоков данных в сети [4, 5].

Функциональная модель сети. Топологию сети будем представлять в виде графа $\Gamma = (V, U)$, имеющего ЯП-форму. Вдоль дуг $u \in U$ графа Γ расположены линии (каналы) передачи данных. В узлах $l \in V$, $l = 1, 2, \ldots, n$, размещены «источники» и «стоки» передаваемых потоков данных. Эти множества разбиты на подмножества V_h и U_h ярусов узлов и дуг, соединяющих V_h, V_{h+1}. Будем считать двудольные графы $\Gamma_h = (V_h \cup V_{h+1}, U_h)$, $h = 1, 2, \ldots, H-1$, полными.

Доставка данных для каждой задачи с номером $k = 1, 2, \ldots, K$ осуществляется по маршрутам графа сети L_j^k, соединяющим узлы, в которых размещены задания k-го графа алгоритма. Максимальные по включению маршруты, далее называемые k-цепями, определяют продолжительность решения k-й задачи. Маршруты L_j^k могут строиться одновременно с решением задачи о назначениях, в которой выбираются сетевые узлы для исполнения соответствующих заданий k-й задачи.

Эффективная и равновесная маршрутизация сети может быть построена с помощью параллельного алгоритма [4, 5]. ЯП-форма графа сети существенно упрощает его реализацию: диспетчеризация заданий и маршрутизация потоков данных могут проводиться одновременно, распространяясь от яруса к ярусу.

С целью поиска эффективного решения этих задач вводят метрики сети, которые локально представляют оптимизируемые критерии качества системы. Пусть для определенности векторный критерий качества системы составлен из продолжительностей T_k, $k = 1, 2, \ldots, K$, решения всех типов задач в сети. Для вычисления и оптимизации этого векторного критерия введем функции задержек $f_l(z_l)$, значения которых зависят от длин очередей z_l из активных заданий в узлах сети $l = 1, 2, \ldots, n$. Задержка — это суммарное время пребывания активного задания в l-м узле, включая его выполнение.

Длину ρ_f маршрута L (метрику сети) определим как сумму задержек:

$$\rho_f(L, z) = \sum_{l \in L} f_l(z_l), \tag{1}$$

где $\mathbf{z} = (z_1, z_2, \ldots, z_l)$.

В выражении (1) не учитывается время передачи по линиям связи. Соответствующую задержку обычно рассматривают в качестве метрики при решении задачи маршрутизации [4].

Теперь можно вычислить $T_k = T_k(\mathbf{z})$ — время решения каждой k-й задачи, как наибольшую длину среди всех максимальных k-цепей в сетевой метрике (1).

Определение функций задержек $f_l(z_l)$, $l \in V$, составляет самостоятельную задачу. Во всяком случае, они неотрицательны, монотонно и неограниченно возрастают при увеличении числа заданий в узле. Непрерывно продолжим функции задержек на множество всех вещественных неотрицательных чисел и будем считать, что $f_l(0) = 0$.

Длины $z_l \geq 0$ и подчиняются балансовым соотношениям

$$\sum_{l \in V_h} z_l = z_{\mathrm{a}h}, \quad h = 1, 2, \ldots, H, \tag{2}$$

означающим постоянство числа $z_{\mathrm{a}h}$ активных заданий, распределяемых среди процессоров любого h-го яруса сети. Предполагается, что в момент принятия решения о выборе вектора $\mathbf{z}$ известен вектор $\mathbf{z}_\mathrm{a} = (z_{\mathrm{a}1}, z_{\mathrm{a}2}, \ldots, z_{\mathrm{a}H})$.

Векторная задача о назначениях. Рассмотрим задачу с векторным критерием быстродействия

$$\mathbf{T}(\mathbf{z}),\ \mathbf{T} = (T_1, T_2, \ldots, T_K),$$

который требуется, по возможности, минимизировать:

$$\mathbf{T}(\mathbf{z}) \to \min, \tag{3}$$

путем выбора неотрицательного вектора $\mathbf{z}$, удовлетворяющего соотношению (2) и получаемого в результате допустимого проецирования $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_K)$ графов алгоритмов в граф сети (см. выше).

Для уточнения постановки векторной задачи с критерием (3) воспользуемся известными принципами оптимальности, изложенными в работе [6]. Напомним, что эффективным по Парето решением многокритериальной задачи (3) на минимум называется такой вектор $\mathbf{z}^*$, что не существует вектора $\mathbf{z}$, при котором строгими являются некоторые неравенства в следующей системе:

$$\mathbf{T}_k(\mathbf{z}) \le \mathbf{T}_k(\mathbf{z}^*), \quad k = 1, 2, \ldots, K. \tag{4}$$

Каждую k-ю задачу о назначениях с оптимизируемым критерием $\mathbf{T}_k$ можно решать «независимо» от других в рамках общей итерационной схемы. Тогда отображение $\boldsymbol{\pi}$ получается в результате поочередного выполнения проецирований π_k графов задач ($k = 1, 2, \ldots, K$). Распределение всех заданий по процессорам дает вектор $\mathbf{z}$, равный сумме векторов $\mathbf{z}^k$ — решений всех «скалярных» задач о назначениях с номерами $k = 1, 2, \ldots, K$. При этом подходе на каждой итерации, на которой происходит решение k-й задачи о назначениях, в балансовых соотношениях (2) варьируется лишь слагаемое $\mathbf{z}^k$.

В пределе (если он существует) получаем решение векторной задачи о назначениях. Принцип оптимальности, характеризующий устойчивость решения, заключается в определении вектора $\mathbf{z}^*$, обладающего свойством

$$\mathbf{T}_k(\mathbf{z}^* \,\|\, \mathbf{z}^k) \le \mathbf{T}_k(\mathbf{z}^*), \quad k = 1, 2, \ldots, K. \tag{5}$$

Здесь через $\mathbf{z}^* \,\|\, \mathbf{z}^k$ обозначен вектор, получаемый из $\mathbf{z}^*$ заменой части координат $\mathbf{z}^{*k}$, относящихся к k-й задаче, на произвольные допустимые значения $\mathbf{z}^k$. Выбор $\mathbf{z}^*$, согласно (5), называется ситуацией равновесия по Нэшу.

Оптимальное решение задачи о назначениях. Пусть $\mathbf{z}^*$ отвечает такому проецированию заданий, которое уравнивает задержки $f_l(z_l)$, $l \in V_h$, на каждом ярусе графа сети. (Если все эти функции одинаковы, то сказанное эквивалентно равномерной загруженности всех процессоров каждого яруса сети.) Решение $\mathbf{z}^*$ не всегда точно реализуемо ввиду целочисленности вектора $\mathbf{z}$.

Рассмотрим задачу (3) в непрерывной постановке, т. е. вектор $\mathbf{z}$ непрерывен. С помощью принципа уравнивания Гермейера [6], применяемого при решении минимаксных задач, можно сформулировать следующую теорему.

Теорема. Уравнивающее задержки назначение $\mathbf{z}^*$ является эффективным по Парето и равновесным по Нэшу решением задачи (3).

Как следует из теоремы, в векторной задаче о назначениях принципы оптимальности (4), (5) не противоречат друг другу. Кроме того, монотонные свойства функций задержек позволяют эффективно реализовать уравнивание значений.

Заметим, что полученный результат вполне соответствует применяемому на практике подходу — стремлению к равномерной загруженности используемого оборудования.

СПИСОК ЛИТЕРАТУРЫ

1. Федоров В.В. О многокритериальных задачах в проектировании // Нелинейное моделирование сложных структур. Вычислительный Центр РАН. – 1987. – С. 127–131.
2. Вязгин В.А., Федоров В.В. Математические методы автоматизационного проектирования. – М.: Высш. шк., 1989. – 106 с.
3. Воеводин В.В. Параллельные вычисления. – М.: Наука, 2002. – 609 с.
4. Васильев Н.С. Задача о кратчайших маршрутах в сетях с переменной метрикой // Вестник МГТУ им. Н.Э. Баумана. Сер. Естественные науки. – 2008 – № 1. – С. 70–75.
5. Васильев Н.С. Задача о равновесной маршрутизации транспортной сети // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. – 2009. – № 2. – С. 76–83.
6. Гермейер Ю.Б. Игры с непротивоположными интересами. – М.: Наука, 1976. – 327 с.

Статья поступила в редакцию 28.09.2012